REMARKS

Claims 1-3 and 11-29 are pending in this application.

Applicants have added new claims 28 and 29. The presentation of these claims does not introduce any new matter.

Claim Amendments

As noted above, Applicants have added new claims 28 and 29. These claims correspond to present claim 16. The subject matter defined in these claims is supported, for example, by Paragraph [0003] of the published application (US 2005/0235268 A1).

Rejections Under 35 U.S.C. § 103

Applicants respectfully request reconsideration of the rejection of claims 1-3, 11-17, and 19-27 under 35 U.S.C. § 103(a) as being unpatentable over *Siska* (US 6,263,429) in view of *Benitez* (US 5,815,721). This rejection is respectfully traversed. As explained in further detail below, the combined teachings of the *Siska* and *Benitez* references would not have rendered the claimed subject matter obvious to one having ordinary skill in the art.

Siska discloses a method of compressing programs, wherein a program is examined for sequences of lines of code that are identical or substantially similar. When such a sequence is identified, an uncompressed version of the sequence is stored as a microroutine in a microroutine area, and each occurrence of the sequence in the program is replaced by a microcall to the newly added microroutine. *Siska's* microroutine collection is determined depending on the actual program that is to be compressed. Consequently, as acknowledged by the Examiner, *Siska* does not disclose that the contents of the predefined library have been determined independently from the compiler-generated program code that is to be optimized.

Furthermore, *Siska* does not disclose the claimed feature that "the compiler-generated program code [is] searched for program code fragments that perform the same function as a respective one of the library code fragments." Instead, *Siska* teaches that program code is

searched for sequences of lines of code that appear multiple times and that are identical or substantially similar. This means that *Siska's* search is performed exclusively on the program code that is to be optimized, without referencing any library or collection of microroutines. In fact, it is apparent from Figure 3A that *Siska's* search (step 300) is performed before any microroutine has been identified for inclusion in the collection (step 302). Thus, as *Siska's* search does not access any type of library, *Siska* cannot disclose a search "for program code fragments that perform the same function as a respective one of the library code fragments," as claimed by Applicants.

As noted above, the Examiner acknowledges that *Siska* does not disclose the contents of a predefined library determined independently from compiler-generated program code that is to be optimized. For this feature of the claimed subject matter, the Examiner looks to the *Benitez* reference. *Benitez* discloses a compiler for optimizing a computer program. *Benitez's* compiler builds abstract web representations for the code segments of the computer program. The compiler also maintains a library of abstract web patterns, wherein each abstract web pattern in the library represents an optimized sequence of computer instructions. The compiler compares each abstract web generated from the code segments with the abstract web patterns in its library. If any of the abstract webs match, the compiler replaces the original code segment in the computer program with the optimized sequence of computer instructions corresponding to the matching abstract web pattern.

The Examiner's rejection appears to equate *Siska's* collection of microroutines with *Benitez's* library of abstract web patterns. However, the two disclosures are not analogous to each other, as *Siska's* collection of microroutines is the result of the optimization, while *Benitez's* library of abstract web patterns represents possible starting points for optimization. These two elements are in fact at the opposite ends of the optimization process. As such, one

having ordinary skill in the art would not have been motivated to combine the teachings of *Benitez* with those of *Siska* in the manner suggested by the Examiner.

In the alternative, if *Siska's* collection of microroutines were to be equated with *Benitez's* optimized sequences of computer instructions, i.e., with the results of the optimization steps taught by *Benitez,* the combined teachings would still fail to disclose the claimed subject matter. This is because *Benitez's* optimized sequences of computer instructions clearly depend on the program code that is to be optimized. This is apparent from the fact that each abstract web pattern of *Benitez* matches a number of different program code sequences. *Benitez,* column 4, lines 17-21 and 57-61. Consequently, the optimizing compiler must be able to produce different optimized sequences of computer instructions for a single abstract web pattern, wherein these optimized sequences of computer instructions depend on the program code that is to be optimized.

The web patterns of *Benitez* can be likened to starting points for optimization rules, wherein the optimization rules are built into the compiler and define how certain program code structures can be optimized. This is apparent from the example discussed at column 5, lines 25-41 of *Benitez*. A suitable web pattern may identify program code fragments that have the general form as shown in lines 25-28. The optimizing compiler of *Benitez* performs a pattern matching operation to discover this general structure, and then uses some built-in optimization rule to generate optimized program code that corresponds to the optimized form shown in lines 37-41. As such, the entire optimized program code (including the program portion denoted by the three dots, which is translated without any recourse to an optimization rule) depends on the program code that is to be optimized.

This is also apparent from *Benitez's* disclosure at column 4, lines 62-65, which states: "If a match is found, then the code can be removed and replaced with a straight-line code sequence that matches the semantics of the original code without the need to explicitly

iterate." Again, the straight-line code that is the result of the optimization needs to match the semantics of the original code and therefore necessarily depends on the program code that is to be optimized. In sum, the optimized sequences of computer instructions of *Benitez* are not "determined independently from the compiler-generated program code that is to be optimized," as claimed by Applicants.

Furthermore, the presently claimed subject matter cannot be considered obvious since an attempt to modify the teachings of *Siska* with the teachings of *Benitez* would fundamentally alter *Siska's* principle of operation. As noted in MPEP § 2143.01, the combined teachings of prior art references are not sufficient to render claims *prima facie* obvious if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified. In the instant case, *Siska's* principle of operation is to determine identical or substantially similar sequences of lines of code within a program, and to use these sequences as microroutines to populate the microroutine area, as is repeatedly stated in *Siska's* specification.

Therefore, the emphasis of *Siska* is to increase the number and size of possible optimizations. For example, *Siska* states in column 1, lines 55-57 that a prior art approach has the disadvantage that "the number of available patterns in a given program is often reduced." As a further example, *Siska* states at column 7, lines 6-10:

> The compression technique of the present invention suitably provides for the compression of significantly greater portions of programs than techniques currently known in the art because the present invention identifies and compresses a greater number of sequences of lines of code within programs.

For the reasons stated above, *Benitez* does not disclose a predefined library analogous to *Siska's* microroutine collection. However, even if such a feature could be found in a reference, then replacing *Siska's* customized collection of microroutines by a predefined library would actually reduce the number and size of possible optimizations because the predefined library would generally contain fewer and shorter matching sequences than the

custom-made collection of *Siska*. Thus, using a predefined library in *Siska* would fundamentally alter *Siska's* principle of operation.

As set forth in Paragraph [0007] of the published application (US 2005/0235268 A1), it is a surprising result of the presently claimed subject matter that a considerable reduction of the size of the program code that is provided for storage in the first memory area can be achieved with the predefined library of the presently claimed subject matter, even if the contents of the predefined library have been determined independently from the compiler-generated program code that is to be optimized. This reduction will usually be less than what would be achievable by a custom-made collection as in *Siska*, but it will still be surprisingly high.

Before the present invention was made, the ordinarily skilled person would not have thought that any meaningful code size savings were possible with a predefined library of a realistic size. It is one of the merits of the present invention to have overcome this prejudice of the ordinarily skilled person.

For at least the foregoing reasons, independent claims 1 and 17 are patentable under 35 U.S.C. § 103(a) over the combined teachings of *Siska* and *Benitez*. Likewise, dependent claims 2-3, 11-16, and 19-27 are allowable by virtue of their dependency from either independent claim 1 or independent claim 17.

Furthermore, with respect to dependent claims 16, 28, and 29, these claims recite that the first memory area occupies more chip area per memory cell in the portable data carrier than is occupied by the second memory area. A typical (but non-limiting) example would be that the first memory area for holding the optimized program code is located in EEPROM, and the second memory area for holding the predefined library is located in mask-programmable ROM.

In the present Office Action, the Examiner refers to column 13, lines 10-30 of *Siska* as allegedly disclosing the above-noted feature. However, this portion of *Siska* merely states that programs for embedded processors may exist in either ROM or RAM. In particular, lines 11-20 teach that, when the compressed program is stored in ROM, then the microroutines are also stored in ROM. On the other hand, lines 21-37 indicate that both the compressed program and the microroutines may be loaded into RAM. In both cases, both the compressed program and the microroutines are located in the same type of memory, which clearly will occupy the same chip area per memory cell, contrary to what is defined in dependent claims 16, 28, and 29.

Furthermore, Figure 2 of *Siska* shows that the microcode area 216 is located in cache memory 206, while the compressed program 210 is located in non-cache memory 208. According to generally used terminology, a cache memory is a fast and volatile memory, i.e., RAM, which occupies a maximum amount of chip area per memory cell, as compared to any other memory types. This disclosure of *Siska* directly contradicts the subject matter defined in dependent claims 16, 28, and 29, which recite that the second memory area (which holds the predefined library) requires less chip area per memory cell than the first memory area.

Applicants respectfully request reconsideration of the rejection of claim 18 under 35 U.S.C. § 103(a) as being unpatentable over *Siska* in view of *Benitez,* further in view of *Wilkinson* (US 2008/0115117 A1). The deficiencies of the prior art with respect to independent claim 17, from which claim 18 depends, have been discussed above. The teachings of the *Wilkinson* reference do not cure the deficiencies of *Siska* and *Benitez* relative to the subject matter defined in independent claim 17. As such, claim 18 is believed to patentable over the cited prior art for at least the reason that this claim depends from independent claim 17.

Conclusion

In view of the foregoing, Applicants respectfully request reconsideration and reexamination of claims 1-3 and 11-27, as well as examination of new claims 28 and 29, and submit that these claims are in condition for allowance. Accordingly, a notice of allowance is respectfully requested. In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at **(408) 749-6902**. If any additional fees are due in connection with the filing of this paper, then the Commissioner is authorized to charge such fees to Deposit Account No. 50-0805 (Order No. WACHP002).

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, L.L.P.

/Peter B. Martine/

Peter B. Martine
Reg. No. 32,043

710 Lakeway Drive, Suite 200
Sunnyvale, California 94085
Customer Number 25920